Les Cahiers du GERAD ISSN: 0711–2440

On a Reduction of the Interval Coloring Problem to a Series of Bandwidth Coloring Problems

M. Bouchard, M. Čangalović,
A. Hertz

G–2007–69

September 2007

On a Reduction of the Interval Coloring Problem to a Series of Bandwidth Coloring Problems

Mathieu Bouchard, Alain Hertz

GERAD and Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal
Montréal (Québec) Canada, H3C 3A7
{mathieu.bouchard, alain hertz}@gerad.ca

Mirjana Čangalović

Faculty of Organizational Sciences
University of Belgrade
Belgrade, Serbia
canga@fon.bg.ac.yu

September 2007

Les Cahiers du GERAD

G–2007–69

Abstract

Given a graph $G = (V, E)$ with strictly positive integer weights ω_i on the vertices $i \in V$, an interval coloring of G is a function I that assigns an interval $I(i)$ of ω_i consecutive integers (called colors) to each vertex $i \in V$ so that $I(i) \cap I(j) = \emptyset$ for all edges $\{i, j\} \in E$. The interval coloring problem is to determine an interval coloring that uses as few colors as possible. Assuming that a strictly positive integer weight δ_{ij} is associated with each edge $\{i, j\} \in E$, a bandwidth coloring of G is a function c that assigns an integer (called a color) to each vertex $i \in V$ so that $|c(i) - c(j)| \geq \delta_{ij}$ for all edges $\{i, j\} \in E$. The bandwidth coloring problem is to determine a bandwidth coloring with minimum difference between the largest and the smallest colors used. We prove that an optimal solution of the interval coloring problem can be obtained by solving a series of bandwidth coloring problems. Computational experiments demonstrate that such a reduction can help to solve larger instances or to obtain better upper bounds on the optimal solution value of the interval coloring problem.

Résumé

Étant donné un graphe $G = (V, E)$ avec des poids entiers strictement positifs ω_i sur les sommets $i \in V$, une coloration par intervalles de G est une fonction I qui attribue un intervalle $I(i)$ comportant ω_i entiers consécutifs (appelés des couleurs) à chaque sommet $i \in V$ de telle sorte que $I(i) \cap I(j) = \emptyset$ pour toute arête $\{i, j\} \in E$. Le problème de la coloration par intervalles consiste à déterminer une coloration par intervalles qui utilise aussi peu de couleurs que possible. En supposant qu'un entier strictement positif δ_{ij} est associé à chaque arête $\{i, j\} \in E$, une coloration par bandes de G est une fonction c qui attribue un entier (appelé une couleur) à chaque sommet $i \in V$ de telle sorte que $|c(i) - c(j)| \geq \delta_{ij}$ pour toute arête $\{i, j\} \in E$. Le problème de la coloration par bandes consiste à déterminer une coloration par bandes qui minimise la différence entre la plus grande et la plus petite couleurs utilisées. Nous démontrons qu'une solution optimale du problème de la coloration par intervalles peut être obtenue en résolvant une série de problèmes de coloration par bandes. Des expériences numériques démontrent qu'une telle réduction peut aider à résoudre des instances plus grandes ou à obtenir de meilleures bornes supérieures sur la valeur optimale du problème de la coloration par intervalles.

1 Introduction

Given a graph $G = (V, E)$ with vertex set V and edge set E, the graph coloring problem is to assign a color to each vertex so that no two adjacent vertices have the same color and the total number of different colors is minimized. We consider two generalizations of this problem, namely the bandwidth and the interval coloring problems.

Assuming that a strictly positive integer weight δ_{ij} is associated with each edge $\{i, j\} \in E$, a bandwidth coloring of G is a function c that assigns an integer (called a color) to each vertex $i \in V$ so that $|c(i) - c(j)| \geq \delta_{ij}$ for all edges $\{i, j\} \in E$. Denoting $max(c) = \max_{i \in V} c(i)$ and $min(c) = \min_{i \in V} c(i)$ the largest and smallest colors used in c, the *span* of a bandwidth coloring c is defined as $span(c) = max(c) - min(c) + 1$. The bandwidth coloring problem is to determine a bandwidth coloring with minimum span. The graph coloring problem is a special case of the bandwidth coloring problem with $\delta_{ij} = 1$ for all edges $\{i, j\} \in E$.

The second generalization of the graph coloring problem assumes that a strictly positive integer weight ω_i is associated with each vertex $i \in V$. An interval coloring of G is a function I that assigns an interval $I(i)$ of ω_i consecutive integers (called colors) to each vertex $i \in V$ so that $I(i) \cap I(j) = \emptyset$ for all edges $\{i, j\} \in E$. The interval coloring problem is to determine an interval coloring that uses as few colors as possible. The special case with $\omega_i = 1$ for all vertices $i \in V$ is equivalent to the graph coloring problem.

Since the graph coloring problem is NP-hard [4], both the bandwidth and the interval coloring problems are NP-hard too. In 2002, Prestwich [5] has described an exact algorithm for solving the bandwidth coloring problem, while an exact algorithm for the interval coloring problem was proposed by Čangalović and Schreuder [2] in 1991.

The aim of this paper is to show that an optimal solution of the interval coloring problem can be obtained by solving a series of bandwidth coloring problems. This reduction is described in the next section with a proof of its validness. Section 3 will be devoted to computational experiments, where we compare the CPU time needed to solve the interval coloring problem using the algorithm in [2] with the total CPU time needed to solve the series of bandwidth coloring problems using the algorithm in [5].

2 The proposed reduction

2.1 Preliminary observations

The main idea of our reduction is based on the simple observation that if two intervals of length ℓ_1 and ℓ_2 are not intersecting, then the distance between their centers is at least $\frac{\ell_1+\ell_2}{2}$. Proposition 1 uses this relation to show the one-to-one correspondence between the bandwidth colorings and the interval colorings of G when all vertex weights are even.

Proposition 1 *Let $G = (V, E)$ be a graph with even weights ω_i on the vertices $i \in V$ and with weights $\delta_{ij} = \frac{\omega_i+\omega_j}{2}$ on the edges $\{i, j\} \in E$. Then c is a bandwidth coloring of G if and only if the intervals $I(i) = \{c(i) - \frac{\omega_i}{2}, \cdots, c(i) + \frac{\omega_i}{2} - 1\}$ define an interval coloring of G.*

Proof. Let c be a bandwidth coloring of G. The intervals $I(i) = \{c(i)-\frac{\omega_i}{2}, \cdots, c(i)+\frac{\omega_i}{2}-1\}$ contain exactly ω_i consecutive integers. To prove that they define an interval coloring of G, we show that $I(i) \cap I(j) = \emptyset$ for all edges $\{i, j\} \in E$. So consider any edge $\{i, j\} \in E$, and assume, without loss of generality, that $c(i) < c(j)$. Then $c(j) \geq c(i) + \frac{\omega_i+\omega_j}{2}$, and the intervals $I(i)$ and $I(j)$ do not intersect since

$$c(j) - \frac{\omega_j}{2} \geq c(i) + \frac{\omega_i + \omega_j}{2} - \frac{\omega_j}{2} = c(i) + \frac{\omega_i}{2}.$$

Conversely, let I be an interval coloring of G and let $min_I(i)$ denote the smallest color in an interval $I(i)$. Consider any edge $\{i, j\} \in E$, and assume, without loss of generality, that $min_I(i) < min_I(j)$. Then $min_I(j) \geq min_I(i) + \omega_i$, which means that

$$c(j) - c(i) = min_I(j) + \frac{\omega_j}{2} - min_I(i) - \frac{\omega_i}{2} \geq \frac{\omega_j + \omega_i}{2} = \delta_{ij}.$$

Hence, c is a bandwidth coloring of G. □

Given a graph $G = (V, E)$ with even weights ω_i on the vertices $i \in V$ and with weights $\delta_{ij} = \frac{\omega_i+\omega_j}{2}$ on the edges $\{i, j\} \in E$, Proposition 1 demonstrates that there is a one-to-one correspondence between the bandwidth and the interval colorings of G. Notice however that an optimal bandwidth coloring of G does not necessarily correspond to an optimal interval coloring, and vice versa. Consider for example, the graph in Figure 1(a), where the numbers into boxes correspond to weights. An optimal bandwidth coloring c of G with $span(c) = 5$ is represented in Figure 1(b). The corresponding interval coloring shown in Figure 1(c) uses 9 colors, which is not optimal. An optimal interval coloring of G with 8 different colors is represented in Figure 1(d). The corresponding bandwidth coloring has span 6, as shown in Figure 1(e).

If the weights on the vertices are not all even, then the intervals in Proposition 1 contain non integer values, and the weights δ_{ij} on the edges are possibly non integer. We will show in Section 2.2.2 how to handle graphs having possibly odd vertex weights.

2.2 The algorithm and its validness

In this section, we describe a procedure that determines an optimal interval coloring by solving a series of bandwidth coloring problems. We first consider graphs with even vertex weights, and we then extend the approach to graphs with general vertex weights. In what follows, we say that an interval coloring I is *compact* if it uses all colors in $\{\min_{i\in V} min_I(i), \cdots, \max_{i\in V}(min_I(i) + \omega_i) - 1\}$.

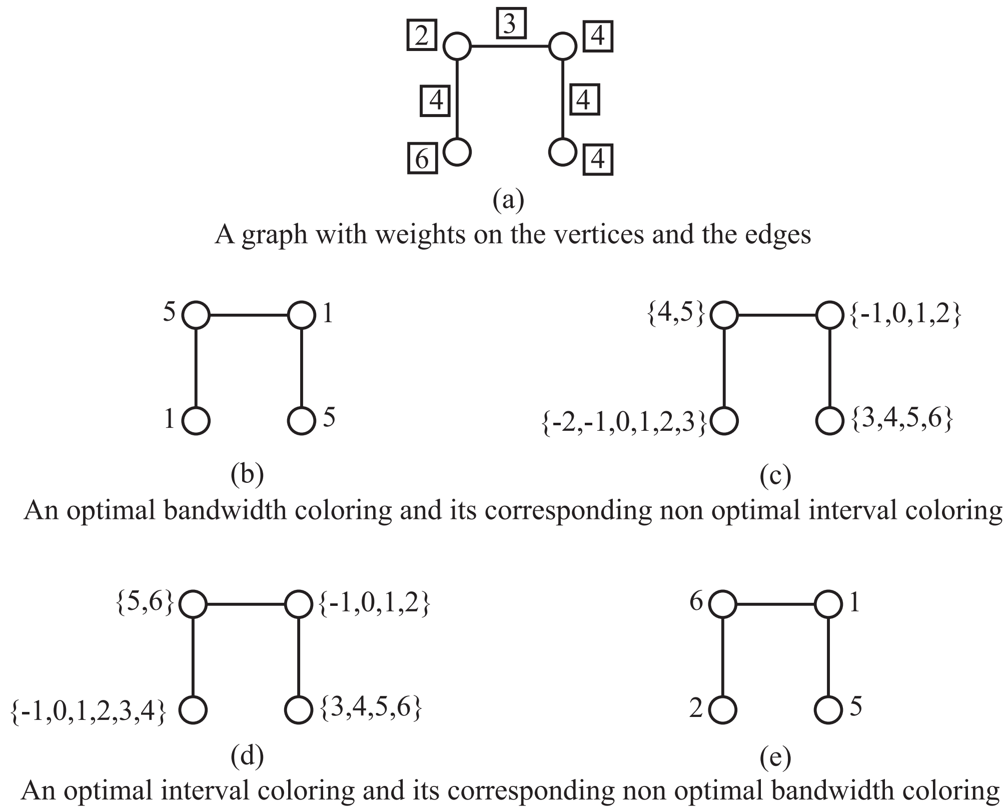

Figure 1: Non correspondence between optimal bandwidth and interval colorings.

2.2.1 Graphs with even vertex weights

Let $G = (V, E)$ be a graph with even weights ω_i on the vertices $i \in V$, and let Δ be any positive integer. We denote G_Δ the edge-weighted graph obtained from G by adding two adjacent vertices α and β linked to all vertices in V, and by setting:

- $\delta_{ij} = \frac{\omega_i + \omega_j}{2}$ for all edges $\{i, j\} \in E$,
- $\delta_{\alpha i} = \delta_{\beta i} = \frac{\omega_i}{2}$ for all vertices $i \in V$,
- $\delta_{\alpha\beta} = \Delta$.

Let $\chi_{int}(G)$ and $\chi_b(G)$ denote the optimal values of the interval and the bandwidth coloring problems on a graph G.

Proposition 2 *If $\Delta \leq \chi_{int}(G)$, then $\chi_{int}(G) \geq \chi_b(G_\Delta) - 1$.*

Proof. Consider the graph G_Δ for an integer $\Delta \leq \chi_{int}(G)$, and let I be a compact optimal interval coloring of G. Define the colors $c_\Delta(i)$ on the vertices of G_Δ as follows:

- $c_\Delta(i) = min_I(i) + \frac{\omega_i}{2}$ for all $i \in V$,
- $c_\Delta(\alpha) = \min_{i\in V} min_I(i)$,
- $c_\Delta(\beta) = \max_{i\in V}(min_I(i) + \omega_i)$.

Since I uses $\max_{i\in V}(min_I(i) + \omega_i) - \min_{i\in V} min_I(i)$ different colors, we have $c_\Delta(\beta) - c_\Delta(\alpha) = \chi_{int}(G)$. Moreover, the colors $c_\Delta(i)$ define a bandwidth coloring of G_Δ. Indeed,

- $|c_\Delta(j) - c_\Delta(i)| \geq \delta_{ij}$ for all edges $\{i,j\} \in E$, as shown in Proposition 1,
- $|c_\Delta(i) - c_\Delta(\alpha)| = min_I(i) + \frac{\omega_i}{2} - \min_{j\in V} min_I(j) \geq \frac{\omega_i}{2} = \delta_{\alpha i}\ \forall i \in V$,
- $|c_\Delta(\beta) - c_\Delta(i)| = \max_{j\in V}(min_I(j) + \omega_j) - min_I(i) - \frac{\omega_i}{2} \geq \frac{\omega_i}{2} = \delta_{\beta i}\ \forall i \in V$,
- $|c_\Delta(\beta) - c_\Delta(\alpha)| = \chi_{int}(G) \geq \Delta = \delta_{\alpha\beta}$.

Since $\chi_{int}(G) = c_\Delta(\beta) - c_\Delta(\alpha) = span(c_\Delta) - 1$, we conclude that $\chi_{int}(G) \geq \chi_b(G_\Delta) - 1$. □

Corollary 1 *Let c_Δ be an optimal bandwidth coloring of G_Δ. If $\Delta \leq \chi_{int}(G)$ and $|c_\Delta(\beta) - c_\Delta(\alpha)| = span(c_\Delta) - 1$ then $\chi_{int}(G) = \chi_b(G_\Delta) - 1$ and the intervals $I(i) = \{c_\Delta(i) - \frac{\omega_i}{2}, \cdots, c_\Delta(i) + \frac{\omega_i}{2} - 1\}$ define a compact optimal interval coloring of G.*

Proof. Without loss of generality, assume $c_\Delta(\alpha) \leq c_\Delta(\beta)$. Since $span(c_\Delta) = c_\Delta(\beta) - c_\Delta(\alpha) + 1$, we have $c_\Delta(\alpha) \leq c_\Delta(i) - \frac{\omega_i}{2}$ and $c_\Delta(\beta) \geq c_\Delta(i) + \frac{\omega_i}{2}$ for all vertices $i \in V$. As shown in Proposition 1, the intervals $I(i) = \{c_\Delta(i) - \frac{\omega_i}{2}, \cdots, c_\Delta(i) + \frac{\omega_i}{2} - 1\}$ define an interval coloring of G. Such a coloring uses at most $\max_{i\in V}(min_I(i) + \omega_i) - \min_{i\in V} min_I(i) \leq c_\Delta(\beta) - c_\Delta(\alpha)$ different colors, which means that

$$\begin{aligned}\chi_{int}(G) &\leq c_\Delta(\beta) - c_\Delta(\alpha)\\ &= span(c_\Delta) - 1 = \chi_b(G_\Delta) - 1\\ &\leq \chi_{int}(G) \text{ (from Proposition 2).}\end{aligned}$$

Hence $\chi_{int}(G) = \chi_b(G_\Delta) - 1$ and the interval coloring I is optimal. The above inequalities also imply that $\chi_{int}(G) = \max_{i\in V}(min_I(i) + \omega_i) - \min_{i\in V} min_I(i)$, which means that I is compact. □

The proposed algorithm for determining an optimal interval coloring of a graph G with even vertex weights by solving a series of bandwidth coloring problems is described in Figure 2. It consists in determining optimal bandwidth colorings c_{Δ_k} in graphs G_{Δ_k} for various values of Δ_k, until $|c_\Delta(\beta) - c_\Delta(\alpha)| = span(c_{\Delta_k}) - 1$.

Theorem 1 *The EvenReduction algorithm is finite and the intervals $I(i)$ produced as output define a compact optimal interval coloring of G.*

Algorithm EvenReduction

Input A graph $G = (V, E)$ with even weights ω_i on the vertices $i \in V$;

Output An optimal interval coloring I of G;

(1) Set $\Delta_1 \leftarrow 0$ and $k \leftarrow 1$;

(2) Determine an optimal bandwidth coloring c_{Δ_k} of G_{Δ_k};
Set $\Delta_{k+1} \leftarrow span(c_{\Delta_k}) - 1$;

(3) If $|c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha)| < \Delta_{k+1}$ then set $k \leftarrow k + 1$, and go to (2);
Else set $I(i) = \{c_{\Delta_k}(i) - \frac{\omega_i}{2}, \cdots, c_{\Delta_k}(i) + \frac{\omega_i}{2} - 1\}$ for all $i \in V$ and STOP.

Figure 2: The proposed algorithm for graphs with even vertex weights.

Proof. We first prove that $\Delta_k \leq \chi_{int}(G)$ at each iteration k. This is obviously true for $k = 1$ since $\Delta_1 = 0 < \chi_{int}(G)$. So assume $\Delta_k \leq \chi_{int}(G)$ and $|c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha)| < span(c_{\Delta_k}) - 1$. Then

$$\Delta_{k+1} = span(c_{\Delta_k}) - 1 = \chi(G_{\Delta_k}) - 1 \leq \chi_{int}(G)$$

the last inequality being valid according to Proposition 2. Now, since

$$\Delta_{k+1} = span(c_{\Delta_k}) - 1 > |c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha)| \geq \Delta_k$$

we know that the algorithm is finite. Finally, since the algorithm stops with $|c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha)| = span(c_{\Delta_k}) - 1$, it follows from Corollary 1 that the intervals $I(i) = \{c_{\Delta_k}(i) - \frac{\omega_i}{2}, \cdots, c_{\Delta_k}(i) + \frac{\omega_i}{2} - 1\}$ define a compact optimal interval coloring of G. □

Let k^* denote the final value of the iteration counter in the EvenReduction algorithm. It corresponds to the number of bandwidth coloring problems which have to be solved to determine an optimal interval coloring of G. The following result is a direct consequence of the proof of Theorem 1.

Corollary 2 $\Delta_1 < \cdots < \Delta_{k^*} \leq \chi_{int}(G)$.

2.2.2 Graphs with general vertex weights

The EvenReduction procedure cannot be applied to graphs with possibly odd vertex weights. Indeed, if ω_i is odd, then the extreme values $c_{\Delta_k}(i) - \frac{\omega_i}{2}$ and $c_{\Delta_k}(i) + \frac{\omega_i}{2} - 1$ of the intervals $I(i)$ are not integer. Also, if an edge $\{i, j\} \in E$ links two vertices with weights of different parity, then the weight $\delta_{ij} = \frac{\omega_i + \omega_j}{2}$ of $\{i, j\}$ in G_{Δ_k} is not integer. The next Proposition shows how to obtain an optimal interval coloring for graphs with general vertex weights.

Proposition 3 *Let $G = (V, E)$ be a graph with weights ω_i on the vertices $i \in V$, and let G' be the same graph as G except that the vertices have weight $\omega'_i = 2\omega_i$ in G'. Given any compact optimal interval coloring I' of G', the intervals $I(i) = \{\left\lceil \frac{min_{I'}(i)}{2} \right\rceil, \cdots, \left\lceil \frac{min_{I'}(i)}{2} \right\rceil + \omega_i - 1\}$ define a compact optimal interval coloring of G.*

Proof. Since all weights in G' are even, it is proved in [1] that $\chi_{int}(G') = 2\chi_{int}(G)$. Let I' be any compact optimal interval coloring of G' and let u be a vertex with minimum value $min_{I'}(u)$ and v a vertex with maximum value $min_{I'}(v) + \omega'_v$. Then $\chi_{int}(G') = min_{I'}(v) + \omega'_v - min_{I'}(u)$, and since $\chi_{int}(G')$ is even, we know that $min_{I'}(u)$ and $min_{I'}(v)$ have the same parity. Hence,

$$\begin{aligned}
\chi_{int}(G) &= \frac{\chi_{int}(G')}{2} \\
&= \frac{min_{I'}(v) + \omega'_v - min_{I'}(u)}{2} \\
&= \left\lceil \frac{min_{I'}(v)}{2} \right\rceil - \left\lceil \frac{min_{I'}(u)}{2} \right\rceil + \omega_v
\end{aligned}$$

Define the intervals $I(i) = \{\left\lceil \frac{min_{I'}(i)}{2} \right\rceil, \cdots, \left\lceil \frac{min_{I'}(i)}{2} \right\rceil + \omega_i - 1\}$ for every vertex $i \in V$. Since the smallest integer in $\cup_{i \in V} I(i)$ is $\left\lceil \frac{min_{I'}(u)}{2} \right\rceil$ while the largest is $\left\lceil \frac{min_{I'}(v)}{2} \right\rceil + \omega_v - 1$, these intervals use $\left\lceil \frac{min_{I'}(v)}{2} \right\rceil + \omega_v - \left\lceil \frac{min_{I'}(u)}{2} \right\rceil = \chi_{int}(G)$ different colors.

It remains to prove that I is an interval coloring of G. Consider any edge $\{i, j\} \in E$. Since the intervals $I'(i)$ and $I'(j)$ do not intersect, assume without loss of generality that $min_{I'}(i) \geq min_{I'}(j) + \omega'_j$.

- If $min_{I'}(i)$ and $min_{I'}(j)$ have the same parity, then

$$min_I(i) - min_I(j) = \frac{min_{I'}(i) - min_{I'}(j)}{2} \geq \frac{\omega'_j}{2} = \omega_j.$$

- if $min_{I'}(i)$ and $min_{I'}(j)$ have different parity, then $min_{I'}(i) - min_{I'}(j) \geq \omega'_j + 1$, which means that

$$min_I(i) - min_I(j) \geq \frac{min_{I'}(i) - min_{I'}(j) - 1}{2} \geq \frac{\omega'_j}{2} = \omega_j.$$

Hence, $I(i)$ and $I(j)$ do not intersect. □

The above proof together with Theorem 1 demonstrate that the following algorithm determines an optimal interval coloring I of any graph G.

The algorithm is illustrated on Figure 4, where the numbers into boxes correspond to weights.

Algorithm GeneralReduction

Input A graph $G = (V, E)$ with weights ω_i on the vertices $i \in V$;

Output An optimal interval coloring I of G;

(1) Construct G' from G by multiplying every weight ω_i by 2;

(2) Determine a compact optimal interval coloring I' of G' using EvenReduction;

(3) Set $I(i) = \{\left\lceil \frac{min_{I'}(i)}{2} \right\rceil, \cdots, \left\lceil \frac{min_{I'}(i)}{2} \right\rceil + \omega_i - 1\}$ for all vertices $i \in V$.

Figure 3: The proposed algorithm for graphs with general vertex weights.

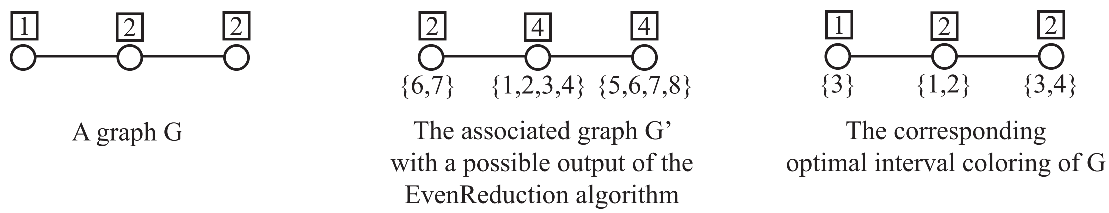

Figure 4: Illustration of the GeneralReduction algorithm.

2.3 Complexity analysis

Let $G = (V, E)$ be a graph with even weights ω_i on the vertices $i \in V$, and let k^* denote the number of bandwidth coloring problems which have to be solved when applying EvenReduction on G. In order to determine an upper bound on k^*, we now associate a weight $\omega_\alpha = \omega_\beta = 0$ to vertices α and β in G_{Δ_k}, and we always assume, without loss of generality, that $c_{\Delta_k}(\alpha) \leq c_{\Delta_k}(\beta)$. For a function f on a set S, let $argmin_{x \in S} f(x)$ and $argmin_{x \in S} f(x)$ respectively denote the set of elements $x \in S$ with minimum and maximum value $f(x)$. For every iteration k of the algorithm, we define

$$\ell_k = \begin{cases} \alpha & \text{if } c_{\Delta_k}(\alpha) < \min_{i \in V} c_{\Delta_k}(i) \\ \text{any vertex in } argmin_{i \in V}(c_{\Delta_k}(i) - \frac{\omega_i}{2}) & \text{otherwise} \end{cases}$$

$$u_k = \begin{cases} \beta & \text{if } c_{\Delta_k}(\beta) > \max_{i \in V} c_{\Delta_k}(i) \\ \text{any vertex in } argmax_{i \in V}(c_{\Delta_k}(i) + \frac{\omega_i}{2}) & \text{otherwise} \end{cases}$$

Proposition 4 *For every iteration $k < k^*$, we have*

(a) $\Delta_{k+1} - \Delta_k \geq \frac{\omega_{\ell_k} + \omega_{u_k}}{2} \geq \chi_{int}(G) - \Delta_{k+1}$,

(b) $\chi_{int}(G) - \Delta_k \geq 2(\chi_{int}(G) - \Delta_{k+1})$,

(c) If $k \geq 3$ *then* $\omega_{\ell_{k-2}} + \omega_{u_{k-2}} > \omega_{\ell_{k-1}} + \omega_{u_{k-1}}$.

Proof. Notice first that

$$c_{\Delta_k}(\alpha) \geq c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2}.$$

Indeed, this is true for $\ell_k = \alpha$ since $c_{\Delta_k}(\alpha) = c_{\Delta_k}(\alpha) + \frac{\omega_\alpha}{2}$. For $\ell_k \neq \alpha$, we have $c_{\Delta_k}(\ell_k) < c_{\Delta_k}(\alpha)$, which means that $c_{\Delta_k}(\alpha) - c_{\Delta_k}(\ell_k) \geq \delta_{\alpha\ell_k} = \frac{\omega_{\ell_k}}{2}$. Similarly,

$$c_{\Delta_k}(\beta) \leq c_{\Delta_k}(u_k) - \frac{\omega_{u_k}}{2}.$$

We therefore have

$$\begin{aligned} \Delta_k &\leq c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha) \\ &\leq c_{\Delta_k}(u_k) - \frac{\omega_{u_k}}{2} - c_{\Delta_k}(\ell_k) - \frac{\omega_{\ell_k}}{2}. \end{aligned} \tag{1}$$

Also, it follows from the definitions of Δ_{k+1} and $span(c_{\Delta_k})$ that

$$\Delta_{k+1} = span(c_{\Delta_k}) - 1 \geq c_{\Delta_k}(u_k) - c_{\Delta_k}(\ell_k). \tag{2}$$

From (1) and (2), we deduce the left inequality in (a) since

$$\begin{aligned} \Delta_{k+1} - \Delta_k &\geq c_{\Delta_k}(u_k) - c_{\Delta_k}(\ell_k) - c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} + c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2} \\ &= \frac{\omega_{\ell_k} + \omega_{u_k}}{2}. \end{aligned}$$

Notice also that

$$c_{\Delta_k}(\ell_k) - \frac{\omega_{\ell_k}}{2} \leq c_{\Delta_k}(i) - \frac{\omega_i}{2} \quad \forall i \in V. \tag{3}$$

Indeed, this inequality follows from the definition of ℓ_k if $\ell_k \neq \alpha$. If $\ell_k = \alpha$ then $c_{\Delta_k}(\alpha) < c_{\Delta_k}(i)$ for all $i \in V$, which means that $c_{\Delta_k}(i) - c_{\Delta_k}(\alpha) \geq \delta_{\alpha i} = \frac{\omega_i}{2}$. Hence, $c_{\Delta_k}(\alpha) - \frac{\omega_\alpha}{2} = c_{\Delta_k}(\alpha) \leq c_{\Delta_k}(i) - \frac{\omega_i}{2}$. Similarly,

$$c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} \geq c_{\Delta_k}(i) + \frac{\omega_i}{2} \quad \forall i \in V. \tag{4}$$

We know from Proposition 1 that the intervals $I(i) = \{c_{\Delta_k}(i) - \frac{\omega_i}{2}, \cdots, c_{\Delta_k}(i) + \frac{\omega_i}{2} - 1\}$ define an interval coloring of G. Hence, from (3) and (4), we have

$$\begin{aligned} \chi_{int}(G) &\leq \max_{i \in V}(c_{\Delta_k}(i) + \frac{\omega_i}{2}) - \min_{i \in V}(c_{\Delta_k}(i) - \frac{\omega_i}{2}) \\ &\leq c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} - c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2}. \end{aligned}$$

So define p as the positive integer such that

$$\chi_{int}(G) = c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} - c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2} - p. \tag{5}$$

From (1) and (5), we have

$$\begin{aligned}\chi_{int}(G) - \Delta_k &\geq c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} - c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2} - p - c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} + c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2} \\ &= \omega_{u_k} + \omega_{\ell_k} - p \qquad (6)\end{aligned}$$

and from (2) and (5), we have

$$\begin{aligned}\chi_{int}(G) - \Delta_{k+1} &\leq c_{\Delta_k}(u_k) + \frac{\omega_{u_k}}{2} - c_{\Delta_k}(\ell_k) + \frac{\omega_{\ell_k}}{2} - p - c_{\Delta_k}(u_k) + c_{\Delta_k}(\ell_k) \\ &= \frac{1}{2}(\omega_{u_k} + \omega_{\ell_k}) - p \\ &\leq \frac{1}{2}(\omega_{u_k} + \omega_{\ell_k} - p). \qquad (7)\end{aligned}$$

This proves the right inequality in (a) as well as (b) since (6) and (7) imply

$$\chi_{int}(G) - \Delta_k \geq 2(\chi_{int}(G) - \Delta_{k+1}).$$

Finally, notice that if $k \geq 3$, then

$$\begin{aligned}\Delta_{k-1} + \frac{\omega_{\ell_{k-2}} + \omega_{u_{k-2}}}{2} &\geq \chi_{int}(G) && \text{(by (a))} \\ &> \Delta_k && \text{(by Corollary 2)} \\ &\geq \Delta_{k-1} + \frac{\omega_{\ell_{k-1}} + \omega_{u_{k-1}}}{2} && \text{(by (a))}\end{aligned}$$

which proves (c). □

In the following, we denote W the number of different weights in G. We now give three upper bounds on the number k^* of bandwidth coloring problems which have to be solved in the EvenReduction algorithm to determine an optimal interval coloring.

Proposition 5 *The three following inequalities define valid upper bounds on k^*:*

(i) $k^* \leq \log_2(\max_{i \in V} w_i) + 3$,

(ii) $k^* \leq 2W + 2$,

(iii) $k^* \leq |V| + 2$.

Proof. To prove (i), notice that it follows from Proposition 4(a) that

$$\chi_{int}(G) - \Delta_2 \leq \frac{1}{2}(\omega_{u_2} + \omega_{\ell_2}) \leq \max_{i \in V} w_i.$$

Hence, by denoting $N = \log_2(\max_{i \in V} w_i) + 1$, we know from Proposition 4(b) that $\chi_{int}(G) - \Delta_{N+2} \leq 0$. It then follows from Corollary 2 that $k^* \leq N + 2$, which means that $k^* \leq \log_2(\max_{i \in V} w_i) + 3$.

To prove (ii), notice first that if $k^* \leq 4$ then $k^* \leq 2W + 2$ since $W \geq 1$. So assume $k^* > 4$. Then for every iteration $k < k^* - 3$ we have

$$\begin{aligned}
max\{\omega_{\ell_k}, \omega_{u_k}\} &\geq \frac{\omega_{\ell_k} + \omega_{u_k}}{2} \\
&\geq \chi_{int}(G) - \Delta_{k+1} && \text{(by Proposition 4(a))} \\
&\geq 2(\chi_{int}(G) - \Delta_{k+2}) && \text{(by Proposition 4(b))} \\
&> 2(\Delta_{k+3} - \Delta_{k+2}) && \text{(by Corollary 2)} \\
&\geq \omega_{\ell_{k+2}} + \omega_{u_{k+2}} && \text{(by Proposition 4(a))} \\
&\geq max\{\omega_{\ell_{k+2}}, \omega_{u_{k+2}}\}
\end{aligned}$$

Since there are W different weights in G while $max\{\omega_{\ell_k}, \omega_{u_k}\} > max\{\omega_{\ell_{k+2}}, \omega_{u_{k+2}}\}$ for all $k < k^* - 3$, we know that at most $2W$ iterations are needed to reach iteration $k^* - 2$, which proves (ii).

To prove (iii), note first that if $k^* \leq 3$, then $k^* \leq |V| + 2$ as $|V| \geq 1$. So assume $k^* > 3$ and define H as the graph with vertex set $V \cup \{\gamma\}$, where γ represents both α and β, and with an edge between two vertices x and y if and only if there is an iteration k with $2 \leq k < k^* - 1$ and $\{x, y\} = \{\ell_k, u_k\}$. To each edge $\{x, y\}$ in H we associate a weight $\omega_x + \omega_y$. Observe that if $\{\ell_k, u_k\} = \{\alpha, \beta\}$, then $span(c_{\Delta_k}) - 1 = c_{\Delta_k}(u_k) - c_{\Delta_k}(\ell_k)$, which means that $k = k^*$. Hence, there is no loop at vertex γ in H. Also, we know from Proposition 4(c) that $\omega_{\ell_{k-1}} + \omega_{u_{k-1}} > \omega_{\ell_k} + \omega_{u_k}$ when $2 \leq k < k^* - 1$, which means that H contains no parallel edge.

Assume H contains a cycle C with edges $\{x_1, x_2\}, \cdots, \{x_{r-1}, x_r\}, \{x_r, x_1\}$ ($r \geq 3$). Without loss of generality, we suppose that $\{x_1, x_2\}$ has maximum weight on C. We then have three iteration values i, j and k such that $\{\ell_i, u_i\} = \{x_1, x_2\}$, $\{\ell_j, u_j\} = \{x_2, x_3\}$, and $\{\ell_k, u_k\} = \{x_1, x_r\}$. Since $\{x_1, x_2\}$ has maximum weight on C, we know from Proposition 4(c) that $i < min\{j, k\}$, and we may suppose without loss of generality that $j < k$. We then have

$$\begin{aligned}
\Delta_{k+1} &\geq \Delta_k + \frac{\omega_{x_1} + \omega_{x_r}}{2} && \text{(by Proposition 4(a))} \\
&\geq \Delta_{j+1} + \frac{\omega_{x_1} + \omega_{x_r}}{2} && \text{(by Corollary 2)} \\
&\geq \Delta_j + \frac{\omega_{x_2} + \omega_{x_3}}{2} + \frac{\omega_{x_1} + \omega_{x_r}}{2} && \text{(by Proposition 4(a))} \\
&\geq \Delta_{i+1} + \frac{\omega_{x_2} + \omega_{x_3}}{2} + \frac{\omega_{x_1} + \omega_{x_r}}{2} && \text{(by Corollary 2)}
\end{aligned}$$

$$
\begin{aligned}
&> \quad \Delta_{i+1} + \frac{\omega_{x_1} + \omega_{x_2}}{2} \\
&\geq \quad \chi_{int}(G) \qquad \text{(by Proposition 4(a))}
\end{aligned}
$$

which contradicts Corollary 2. Hence H has no cycle, which means that it contains at most V edges. As a consequence, $k^* \leq |V| + 2$. □

Notice that the three bounds are sharp as illustrated in Figure 5 with a graph having two vertices with weight 2. We have represented the successive graphs G_{Δ_k} with optimal bandwidth colorings c_{Δ_k}. The colors are the numbers close to the vertices while the weights are shown into boxes. Since $span(G_{\Delta_4}) - 1 = c_{\Delta_4}(\beta) - c_{\Delta_4}(\alpha)$, we have $k^* = 4 = \log_2(\max_{i \in V} w_i) + 3 = 2W + 2 = |V| + 2$.

Observe also that the EvenReduction algorithm could have reached the optimal solution in one less iteration since the bandwidth coloring c_{Δ_4} is also optimal for G_{Δ_3}. Also, instead of initializing the EvenReduction algorithm with $\Delta_1 = 0$, one could set Δ_1 equal to any lower bound on $\chi_{int}(G)$. For example, one could determine a maximal (inclusion wise) clique (i.e., a set of pairwise adjacent vertices) in G and set Δ_1 equal to the total weight of this clique. In the above example, the unique maximal clique contains vertices v and w for a total weight of 4. By setting $\Delta_1 = 4$, the EvenReduction algorithm could skip the first three iterations, and would find the optimal interval coloring by solving only one bandwidth coloring problem.

3 Computational experiments

In this section, we report computational experiments in order to compare the CPU time needed to solve the interval coloring problem using the algorithm in [2] with the total CPU time needed to solve the series of bandwidth coloring problems using the algorithm in [5].

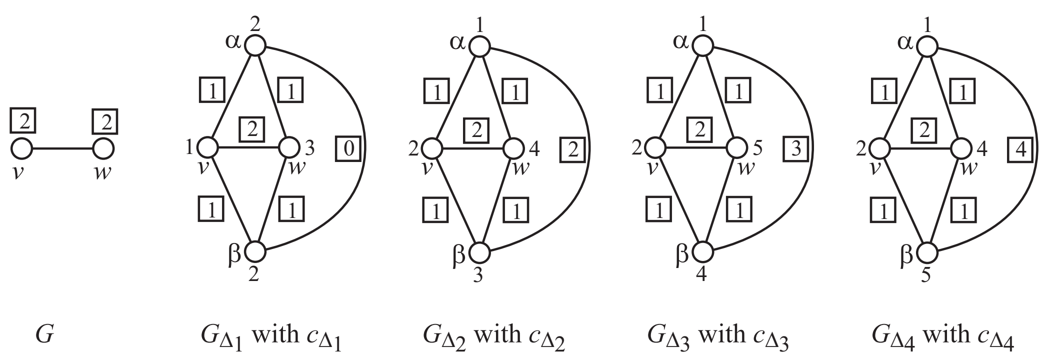

Figure 5: The three upper bounds on k^* are sharp.

We have considered two test sets. The first experiments are performed on random graphs $G_{n,p,q}$ already used in [2] and [3]. Given a positive integer n, a real number $p \in [0,1]$, and a positive real number q, a random graph $G_{n,p,q}$ contains n vertices, all $n(n-1)/2$ ordered pairs of vertices have a probability p of being linked by an edge, and the weight ω_i of a vertex i is generated according to an independent truncated Poisson variable with parameter q, i.e., the probability that $\omega_i = m$ for some positive integer m is set equal to $\frac{q^m}{(e^q-1)m!}$.

The second set of instances used for the experiments are the DIMACS benchmark graphs, which come from various sources. For a detailed description of these instances, the reader can refer to `http://mat.gsia.cmu.edu/COLOR04`.

We call CS-Interval the exact algorithm proposed by Čangalović and Schreuder in [2] for finding an optimal interval coloring. It is based on the Branch-and-Bound principle. An initial lower bound on $\chi_{int}(G)$ is obtained by determining a clique of maximum total weight, using a variation of the algorithm proposed in [6]. Also, an initial upper bound on $\chi_{int}(G)$ is obtained by using the heuristic algorithm proposed in [3]. The exact algorithm proposed by Prestwich in [5] for the bandwidth coloring will be called P-Bandwidth. It combines a local search algorithm with a backtracking technique that uses constraint propagation.

Let LB denote the above mentioned lower bound on $\chi_{int}(G)$, and let G' be the graph obtained from G by multiplying every weight ω_i by 2 (see Step (1) of the GeneralReduction algorithm). Since Δ_1 can be set equal to any lower bound on $\chi_{int}(G') = 2\chi_{int}(G)$, we use $\Delta_1 = 2LB$ in order to reduce the number k^* of bandwidth coloring problems which have to be solved. Notice that if $\chi_{int}(G) = LB$, then $\chi_{int}(G') = \Delta_1$ and it follows from Corollary 2 that $k^* = 1$. Otherwise, if $\chi_{int}(G) > LB$, denote $N = \log_2(\chi_{int}(G') - \Delta_1) + 1$. Using the same arguments as in the proof of Proposition 5 we get $k^* \leq N+1 = \log_2(\chi_{int}(G) - LB) + 1$.

For all our tests, we have considered a time limit of one hour on an AMD Opteron(tm) 285/2.6 GHz Processor. It can happen that this time is not sufficient to reach an optimal solution or to prove its optimality. When the CS-Interval algorithm has to be stopped before its end, we report the number of colors used in the best interval coloring encountered during the search, which is possibly not optimal. If the GeneralReduction algorithm has to be stopped while running the P-bandwidth algorithm, we translate the current best bandwidth coloring into an interval coloring using the correspondance described in Proposition 1 and report its number of colors.

Since it may happen that the P-Bandwidth algorithm needs more than one hour during its first run (i.e., when applied on G'_{Δ_1}), we have also tested a version of the GeneralReduction algorithm where each run of the P-Bandwidth algorithm is stopped after at most 10 million backtracks, the overall time limit being however kept equal to one hour. This allows to run the P-Bandwidth algorithm on several graphs G'_{Δ_k} with increasing Δ_k. The resulting solution of the GeneralReduction algorithm then corresponds to an upper bound on the optimal value when at least one run of the P-Bandwidth algorithm is stopped before its end.

Table 1 contains the results for the graphs $G_{n,p,q}$ with $n = 20, 30, 40, 50$, $p = 0.3, 0.4, 0.5$, and $q = 1, 5, 10$. Three graphs have been generated for each such triplet (n, p, q), which gives a total of 108 instances. For each type of graph, we indicate the average value of the largest vertex weight (column labeled "max ω") and the average number of different vertex weights (column labeled "W"). We also report the average value of the lower and upper bounds (columns labeled "LB" and "UB") mentioned above. For each algorithm, we report the average number of colors in the best found interval coloring (column "χ_{int}"), the number of graphs among the three tested where optimality could be proved (column "opt"), and the average CPU time in seconds for those runs that needed less than the one hour time limit. For the GeneralReduction algorithm, we also indicate the number k^* of bandwidth coloring problems which had to be solved. The last line contains the average numbers of colors produced by each algorithm.

We can observe (see column "opt") that the CS-Interval algorithm easily finds the optimum value for graphs $G_{n,p,q}$ with a small number of vertices or with a small value of q. This is not always the case for the two versions of the GeneralReduction algorithm. For example, for $n = 20$, $p = 0.5$ and $q = 10$, no instance could be solved to optimality with the GeneralReduction algorithm, while the CS-Interval algorithm solves the three instances with an average of 30 seconds. However, the upper bounds produced by the GeneralReduction algorithm are often optimal values. For example, for $n = 50$, $p = 0.4$, $q = 1$, the CS-Interval algorithm produces three optimal solutions with an average of 13.33 colors, while the GeneralReduction algorithm with the limit of 10 million backtracks produces colorings with the same number of colors, but without any proof of optimality.

The CS-Interval algorithm has however more difficulty to solve larger graphs within the one hour time limit. While the GeneralReduction algorithm also fails in finding optimal solutions, it produces in these cases better upper bounds. For example, for $n = 50$, $p = 0.5$ and $q = 10$, the average upper bound found by the CS-Interval algorithm is 122.33, while solutions with in average 114 colors could be produced using the GeneralReduction algorithm. For the 108 instances of this first experiment, the CS-Interval algorithm produces solutions with an average of 41.08 colors. The GeneralReduction algorithm performs a little bit better since 40.37 colors are obtained with no limit on the number of backtracks in the P-Bandwidth algorithm, and 40.02 colors otherwise.

We observe that the 10 million backtracks limit typically has a positive impact on the output of the GeneralReduction algorithm. For example, for $n = 40$, $p = 0.4$ and $q = 10$, one hour is not sufficient for the P-Bandwidth algorithm to reach an optimal solution on its first run, and the best found solution uses 76.33 colors, in average. When stopping the P-Bandwidth algorithm after at most 10 million backtracks, it is possible to run it iteratively on three different graphs (with increasing Δ_k) and this allows to obtain colorings that use

Table 1: Results for random graphs $G_{n,p,q}$

instance							CS-Interval			GeneralReduction							
										No backtrack limit				≤10 million backtracks			
n	p	q	max ω	W	LB	UB	χ_{int}	opt	CPU	χ_{int}	opt	CPU	k^*	χ_{int}	opt	CPU	k^*
20	0.3	1	3	3	7.00	7.00	7.00	3	0	7.00	3	0	0	7.00	3	0	0
30			4	3	8.67	9.00	9.00	3	0	9.00	2	0	0.67	9.00	2	20	0.67
40			3	3	9.00	11.00	10.00	3	0	10.00	0	-	1.33	10.00	0	72	2.00
50			4	4	11.33	13.00	12.00	3	0	12.33	1	0	1.00	12.00	1	57	1.67
20	0.3	5	10	8	24.67	26.67	26.00	3	0	26.00	2	0	0.67	26.00	2	46	1.00
30			11	10	28.33	31.33	29.00	3	0	29.00	2	1	1.00	29.00	2	63	1.33
40			11	8	31.33	36.67	32.33	3	2	32.67	1	6	1.00	32.33	1	66	1.00
50			10	9	30.33	43.33	36.33	3	1283	38.33	0	-	1.00	36.33	0	279	2.33
20	0.3	10	14	9	37.33	42.00	38.33	3	6	39.33	0	-	1.00	39.33	0	133	1.00
30			16	11	50.67	59.67	55.00	3	7	56.00	0	-	1.00	55.00	0	268	1.33
40			18	12	57.33	68.00	63.00	1	6	61.00	1	61	1.00	60.00	2	239	1.33
50			18	14	60.67	84.33	78.67	0	-	73.67	0	-	1.00	73.00	0	1114	3.00
20	0.4	1	4	4	10.00	10.67	10.00	3	0	10.00	3	0	0.67	10.00	3	0	0.67
30			4	4	12.67	13.33	12.67	3	0	12.67	3	0	0.67	12.67	3	0	0.67
40			3	3	11.00	13.00	11.67	3	0	12.00	1	8	1.00	11.67	1	76	2.00
50			4	4	12.67	14.67	13.33	3	6	13.67	1	548	1.00	13.33	0	82	2.00
20	0.4	5	10	8	27.00	28.00	27.00	3	0	27.00	3	0	0.33	27.00	3	0	0.33
30			10	9	35.00	40.67	36.67	3	226	37.33	0	-	1.00	37.00	1	142	1.67
40			11	10	31.67	41.67	36.33	3	487	38.33	0	-	1.00	37.00	0	254	2.33
50			9	9	38.67	54.67	47.67	0	-	49.33	0	-	1.00	48.67	0	578	3.33
20	0.4	10	15	8	51.67	55.00	53.33	3	0	54.33	1	0	0.67	53.67	1	90	0.67
30			17	11	61.67	75.33	64.67	3	93	65.00	1	40	1.00	65.00	1	247	1.33
40			15	12	59.67	83.67	78.33	0	-	76.33	0	-	1.00	74.00	0	854	3.00
50			18	14	74.33	103.67	102.00	0	-	91.33	0	-	1.00	90.67	0	982	2.67
20	0.5	1	3	3	9.00	9.67	9.67	3	0	9.67	1	0	1.33	9.67	1	32	1.33
30			3	3	12.33	13.00	12.33	3	0	12.33	3	0	0.67	12.33	3	0	0.67
40			3	3	13.00	16.00	14.33	3	0	15.00	0	-	1.00	14.33	0	122	2.33
50			4	4	15.33	20.33	17.67	3	355	18.33	0	-	1.00	18.00	0	196	3.33
20	0.5	5	10	8	34.33	36.67	34.67	2	0	35.33	1	0	0.67	35.00	1	47	0.67
30			10	9	38.33	45.33	40.67	3	85	41.33	1	8	1.00	41.33	0	188	2.00
40			9	9	37.33	50.67	45.33	1	1561	46.00	0	-	1.00	45.33	0	385	3.33
50			9	9	42.00	59.33	55.33	0	-	52.33	0	-	1.00	52.00	0	463	3.33
20	0.5	10	16	8	52.00	58.67	54.00	3	30	54.33	0	-	1.00	54.33	0	183	1.33
30			19	11	68.33	85.33	77.67	1	1419	78.00	0	-	1.00	77.00	0	762	2.33
40			16	12	73.33	111.33	104.67	0	-	95.00	0	-	1.00	96.67	0	1255	4.00
50			19	13	88.00	129.33	122.33	0	-	114.00	0	-	1.00	115.00	0	1669	4.00
average					35.17	44.50	41.08			40.37				40.02			

only 74 colors in average. This is however not always the case as illustrated by the graphs with $n = 50$, $p = 0.5$ and $q = 10$. Indeed, one run of the P-Bandwidth algorithm during one hour produces a coloring with 114 colors in average, while 115 colors are obtained with 4 runs of the P-Bandwidth algorithm with the limit of 10 million backtracks.

Table 2 reports computational experiments on the DIMACS benchmark graphs. These instances have up to 191 vertices and are therefore more challenging. The three first columns contain the name of the instances, their number n of vertices, and their number m of edges. The next columns are similar to those of Table 1 except that column "opt" is replaced with the following rule: when an algorithm does not produce a solution with a proof of optimality, we report the obtained upper bound under column "χ_{int}" with italic characters, while normal characters are used when optimality is proved. CPU times are only given for instances where the GeneralReduction algorithm stopped before the 1 hour time limit. We do not report any result for instances R50_1g, R50_1gb and GEOMx with x = 20, 20a, 20b, 30, 30a, 30b, 40, 40a, 40b, 50, 50a, 60, 70, 70a, 80, 90b, 100a, 110a, 110b, since the lower bound LB and the upper bound UB are equal on these graphs (hence there is no need to use a Branch-and-Bound procedure).

The CS-Interval algorithm is able to solve 9 instances to optimality within the one hour time limit, while a proof of optimality is obtained 13 times for each version of the GeneralReduction algorithm. The GEOM instances seem to be easier for the GeneralReduction algorithm since 11 of the 13 instances are solved to optimality, while the CS-Interval algorithm solves only two of them. A deeper analysis indicates that the 6 instances which are solved to optimality by the CS-Interval algorithm and not by the GeneralReduction algorithm all have $\chi_{int}(G) > LB$, while the 10 instances which are solved to optimality by the GeneralReduction algorithm and not the CS-Interval algorithm all have $\chi_{int}(G) = LB$. Hence the GeneralReduction algorithm seems to be particularly effective in proving optimality when the lower bound equals the optimal value. This is confirmed by the results in Table 1 since there are only 5 triplets (n, p, q) for which we know that $\chi_{int}(G_{n,p,q}) = LB$ (namely $(20, 0.3, 1), (20, 0.4, 1), (30, 0.4, 1), (20, 0.4, 5)$ and $(30, 0.5, 1)$), and the GeneralReduction algorithm is able to obtain a proof of optimality (i.e., opt=3) only on these instances.

We next observe from Table 2 that the CS-Interval algorithm produces solutions with an average of 106.12 colors while, again, the GeneralReduction algorithm performs better in average since 103.44 colors are found with no limit on the number of backtracks in the P-Bandwidth algorithm, and 102.87 colors otherwise. We also notice that if we except four instances (namely queen8_8g, queen9_9g, R100_1g and R50_5g), the number of colors produced by the GeneralReduction algorithm with the limit of 10 million backtracks is never larger than the number of colors obtained with the CS-Interval algorithm, and it is even strictly smaller for 34 of the 52 instances.

Notice also that the GeneralReduction algorithm with the 10 million backtracks limits has required less than one hour CPU for 42 instances. This means that the P-Bandwidth algorithm has produced a coloring c_{Δ_k} with $|c_{\Delta_k}(\beta) - c_{\Delta_k}(\alpha)| = \Delta_{k+1}$ within the one hour time limit. As mentioned above, this does not mean that the resulting solution is optimal since previous runs of the P-Bandwidth algorithm with smaller values of Δ_k have possibly reached the 10 million backtracks limit.

Table 2: Results for DIMACS benchmark graphs

instance							CS-Interval		GeneralReduction					
									No backtrack limit			≤10 million backtracks		
name	n	m	max ω	W	LB	UB	χ_{int}	CPU	χ_{int}	CPU	k^*	χ_{int}	CPU	k^*
DSJC125.1gb	125	736	20	20	67	86	*82*	-	*73*	-	1	*74*	928	2
DSJC125.1g	125	736	5	5	19	23	19	7	19	83	1	19	83	1
DSJC125.5gb	125	3891	20	20	125	246	*246*	-	*239*	-	1	*237*	-	4
DSJC125.5g	125	3891	5	5	40	72	*72*	-	*68*	-	1	*69*	584	3
DSJC125.9gb	125	6961	20	20	425	608	*608*	-	*608*	-	1	*608*	-	1
DSJC125.9g	125	6961	5	5	122	166	*166*	-	*163*	-	1	*164*	-	4
GEOM50b	50	249	3	3	17	18	17	0	17	0	1	17	0	1
GEOM60a	60	339	10	10	65	66	*66*	-	65	56	1	65	56	1
GEOM60b	60	366	3	3	22	23	22	508	22	2	1	22	2	1
GEOM70b	70	488	3	3	22	23	*23*	-	22	12	1	22	12	1
GEOM80a	80	612	10	10	68	72	*72*	-	*69*	-	1	*69*	1065	2
GEOM80b	80	663	3	3	25	26	*26*	-	25	1	1	25	1	1
GEOM90a	90	789	10	10	65	70	*69*	-	65	0	1	65	0	1
GEOM90	90	441	10	10	51	52	*52*	-	51	9	1	51	9	1
GEOM100b	100	1050	3	3	30	31	*31*	-	30	0	1	30	0	1
GEOM100	100	547	10	10	60	61	*61*	-	60	0	1	60	0	1
GEOM110	110	638	10	10	62	67	*67*	-	62	46	1	62	46	1
GEOM120a	120	1434	10	10	93	98	*98*	-	*98*	-	1	*98*	2661	3
GEOM120b	120	1491	3	3	34	35	*35*	-	34	10	1	34	10	1
GEOM120	120	773	10	10	63	68	*68*	-	*67*	-	1	*65*	1645	2
myciel5gb	47	236	20	20	37	65	53	0	*57*	-	1	*53*	838	3
myciel5g	47	236	5	5	10	19	17	1	*19*	-	1	*17*	134	3
myciel6gb	95	755	20	20	39	92	*82*	-	*84*	-	1	*79*	2243	4
myciel6g	95	755	5	5	10	25	20	308	*21*	-	1	*20*	272	3
myciel7gb	191	2360	20	20	40	98	*93*	-	*93*	-	1	*90*	2185	3
myciel7g	191	2360	5	5	10	28	*28*	-	*24*	-	1	*24*	541	4
queen10_10gb	100	1470	20	20	136	159	*159*	-	*147*	-	1	*146*	-	2
queen10_10g	100	1470	5	5	38	43	*40*	-	*39*	-	1	*39*	618	2
queen11_11gb	121	1980	20	19	140	170	*170*	-	*168*	-	1	*165*	2335	3
queen11_11g	121	1980	5	5	41	48	*44*	-	*44*	-	1	*44*	618	2
queen12_12gb	144	2596	20	20	163	192	*192*	-	*179*	-	1	*180*	-	3
queen12_12g	144	2596	5	5	42	52	*49*	-	*49*	-	1	*49*	536	3
queen8_8gb	64	728	20	20	113	120	*117*	-	113	220	1	113	220	1
queen8_8g	64	728	5	5	28	34	*29*	-	*32*	-	1	*30*	276	3
queen9_9gb	81	1056	20	20	135	157	*154*	-	*148*	-	1	*145*	-	2
queen9_9g	81	1056	5	5	35	39	35	615	*36*	-	1	*36*	236	1
R100_1gb	100	509	20	20	56	78	*78*	-	*70*	-	1	*65*	800	2
R100_1g	100	509	5	5	15	19	17	3	*19*	-	1	*18*	132	2
R100_5gb	100	2456	20	20	132	222	*222*	-	*208*	-	1	*211*	-	4
R100_5g	100	2456	5	5	35	58	*58*	-	*54*	-	1	*55*	461	3
R100_9gb	100	4438	20	20	390	512	*512*	-	*512*	-	1	*512*	-	1
R100_9g	100	4438	5	5	108	140	*140*	-	*134*	-	1	*134*	1422	4
R50_5gb	50	612	20	19	98	131	*130*	-	*118*	-	1	*119*	1993	4
R50_5g	50	612	5	5	27	34	*32*	-	*34*	-	1	*33*	338	4
R50_9gb	50	1092	19	18	228	264	*264*	-	*264*	-	1	*264*	-	1
R50_9g	50	1092	5	5	64	73	*73*	-	*69*	-	1	*68*	637	4
R75_1gb	70	251	20	20	53	69	*63*	-	*57*	-	1	*57*	303	1
R75_1g	70	251	5	5	14	19	16	0	*17*	-	1	*16*	151	2
R75_5gb	75	1407	20	20	114	184	*184*	-	*172*	-	1	*171*	2996	4
R75_5g	75	1407	5	5	31	50	*50*	-	*46*	-	1	*46*	386	3
R75_9gb	75	2513	20	20	298	393	*393*	-	*393*	-	1	*393*	-	1
R75_9g	75	2513	5	5	85	104	*104*	-	*102*	-	1	*101*	1150	4
average					81.53	107.73	106.12		103.44			102.87		

4 Conclusion

We have shown that an optimal solution of the interval coloring problem can be obtained by solving a series of bandwidth coloring problems. Since both the interval and the bandwidth coloring problems are NP-hard, such a reduction could appear as useless. However, we have shown that when a lower bound LB on $\chi_{int}(G)$ can be obtained and it has to be proved that $\chi_{int}(G) = LB$, then the proposed reduction is particularly efficient since it allows to solve larger instances than those solved by the exact CS-Interval algorithm. Notice that $k^* = 1$ in these cases, which means that the solution of the interval coloring problem is reduced to the solution of exactly one bandwidth coloring problem.

We have shown that for large instances, where the optimal solution can hardly be obtained in a reasonable amount of time, the proposed reduction helps producing better upper bounds on $\chi_{int}(G)$. An average improvement of more than 3% could be observed on the DIMACS benchmark graphs. Also, we have shown that by fixing a limit on the number of backtracks when solving each bandwidth coloring problem with a Branch-and-Bound algorithm, it is possible to use increasing values of Δ_k and to typically obtain better upper bounds on $\chi_{int}(G)$.

References

[1] M. Čangalović. Some new combinatorial optimization algorithms applied to timetabling problems. Ph.D. thesis, University of Belgrade, Serbia (in Serbian), 1989.

[2] M. Čangalović, J.A.M. Schreuder. Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research*, 51:248–258, 1991.

[3] A.T. Clementson, C.H. Elphick. Approximate colouring algorithms for composite graphs. *Journal of Operational Research Society*, 34/6:503–509, 1983.

[4] M.R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freman and Company, NY, 1979.

[5] S. Prestwich Constrained bandwidth multicoloration neighborhoods. *Proceedings of Computational Symposium on Graph Coloring and its Generalizations*,Ithaca, NY, USA, 126–133, 2002.

[6] T. Sakaki, K. Nakashima, Y. Hattori. Algorithms for finding in the lump both bounds of the chromatic number of a graph. *The Computer Journal*, 19:329–332, 1976.